**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium: Grafika Komputerowa**

04.03.2020

**Temat:** Grafika 2D z u˙zyciem HTML Canvas”

**Wariant:** 11,15

Michał Krzyżowski

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr.1b

**1. Polecenie:**

Zadanie na labolatoriach składało się z dwóch poleceń:

1. Narysować obraz zgodnie z wariantem zadania (Moją figurą była figura numer 11 – uśmiechnięty oval) (używając zarówno standardowe jak i niestandardowe funkcje rysowania).

2. W pliku Lab2Ex2.html program domyślnie rysował szereg kwadratów. Celem było stworzyć narzędzia pozwalające na wykonywanie czynności:

* Czyszczenie
* Dodanie jednego nowego koloru do elementu select, zaimplementować nową funkcje poprzez polecenie doMouseMove.

**2. Wprowadzane dane:**

W programie nie wprowadzałem żadnych danych – w pełni operowałem na przygotowanym szablonie i możliwościach Canvas. W zadaniu pierwszym stworzyłem wymaganą figurę, natomiast w drugim dodałem dodatkowe opcje umożliwiające wybór użytkownikowi figury/ koloru/ opcje wyczyść.

**3. Wykorzystane komendy:**

Komendy wykorzystywane podczas pracy nad projektem to instrukcje: graphics.lineWidth by ustalić grubość linii, graphics.stokeOval by na rysować krawędzie ovalu, graphics.fillStyle by ustawić kolor wypełnienia, graphics.fillOval by wypełnić owal. Podobne instrukcje rysowania i wypełniania były użyte w celu stworzeni kulek czy prostokątów. Używana także była krzywa Beziera do malowania powierzchni nie będących fiugrami, ani liniami prostymi: graphics.bezierCurveTo. Komenda graphics.moveTo aby się przemieszczać narzędzie rysującym. W drugim zadaniu należało skorzystać ze składni javascriptu i htmlu, aby dodać nowe pozycje w <select>, oraz by zrobić uchwyt w JS, poprzez użycie getElementById. Reszta komend używanych była bardzo podobna jak w zadaniu pierwszym dodatkowo zawierając w sobie tworzenie poligonu poprzez graphics.fillPoly.

1. Kod do zadania 1-szego tworzący uśmiechnięty owal.

graphics.lineWidth=3;

graphics.strokeOval(300,300,200,100);

graphics.fillStyle="purple";

graphics.fillOval(300,300,200,100);

graphics.fillStyle="#FFFFFF";

graphics.strokeCircle(250,280,20);

graphics.fillCircle(250,280,19);

graphics.strokeCircle(350,280,20);

graphics.fillCircle(350,280,19);

graphics.fillStyle="black";

graphics.strokeCircle(245,275,5);

graphics.fillCircle(245,275,5);

graphics.strokeCircle(345,275,5);

graphics.fillCircle(345,275,5);

graphics.fillStyle="white";

graphics.strokeCircle(242,275,2);

graphics.fillCircle(242,275,2);

graphics.strokeCircle(342,275,2);

graphics.fillCircle(342,275,2);

graphics.beginPath();

graphics.moveTo(270,330);

graphics.bezierCurveTo(270,360,330,360,330,330);

graphics.bezierCurveTo(330,340,270,340,270,330);

graphics.fillStyle="black";

graphics.fill();

graphics.stroke();

graphics.moveTo(265,330);

graphics.quadraticCurveTo(270,330,275,325);

graphics.moveTo(325,325);

graphics.quadraticCurveTo(330,330,335,330);

graphics.stroke();

graphics.fillStyle="white";

graphics.lineWidth=1;

graphics.fillRect(297,339,3,5);

graphics.strokeRect(296,339,4,6);

graphics.fillRect(300,339,3,5);

graphics.strokeRect(300,339,4,6);

graphics.beginPath();

graphics.moveTo(270,330);

graphics.bezierCurveTo(270,360,330,360,330,330);

graphics.bezierCurveTo(330,340,270,340,270,330);

graphics.fillStyle="black";

graphics.fill();

graphics.stroke();

graphics.moveTo(265,330);

graphics.quadraticCurveTo(270,330,275,325);

graphics.moveTo(325,325);

graphics.quadraticCurveTo(330,330,335,330);

graphics.stroke();

graphics.fillStyle="white";

graphics.lineWidth=1;

graphics.fillRect(297,339,3,5);

graphics.strokeRect(296,339,4,6);

graphics.fillRect(300,339,3,5);

graphics.strokeRect(300,339,4,6);

b) kod źródłowy – zadanie 2 (kod zawiera dodanie koloru, oraz ustawianie bryły.

/\*------------------------------------------------------------\*/

/\* TODO: Add support for more drawing tools. \*/

if ( Math.abs(x-prevX) + Math.abs(y-prevY) < 3 ) {

return; // don't draw squares too close together

}

if (colorChoice == 0) {

graphics.fillStyle = randomColorString();

}

else if (colorChoice == 1) {

graphics.fillStyle = "red";

}

else if (colorChoice == 2) {

graphics.fillStyle = "green";

}

else if (colorChoice == 3) {

graphics.fillStyle = "blue";

}

else if (colorChoice ==4)

{

graphics.fillStyle="cyan";

}

if (shapesChoice == 0){

graphics.fillRect(x-20,y-20,40,40);

graphics.strokeRect(x-20,y-20,40,40);

}

else if(shapesChoice == 1)

{

graphics.fillPoly(x-20\*Math.cos((2\*2\*Math.PI/15)),y-20\*Math.sin((2\*2\*Math.PI/15)),x-20\*Math.cos((2\*2\*Math.PI/15)),y-20\*Math.sin((2\*2\*Math.PI/15)),

x-20\*Math.cos((3\*2\*Math.PI/15)),y-20\*Math.sin((3\*2\*Math.PI/15)),x-20\*Math.cos((4\*2\*Math.PI/15)),y-20\*Math.sin((4\*2\*Math.PI/15)),

x-20\*Math.cos((5\*2\*Math.PI/15)),y-20\*Math.sin((5\*2\*Math.PI/15)),x-20\*Math.cos((6\*2\*Math.PI/15)),y-20\*Math.sin((6\*2\*Math.PI/15)),

x-20\*Math.cos((7\*2\*Math.PI/15)),y-20\*Math.sin((7\*2\*Math.PI/15)),x-20\*Math.cos((8\*2\*Math.PI/15)),y-20\*Math.sin((8\*2\*Math.PI/15)),

x-20\*Math.cos((9\*2\*Math.PI/15)),y-20\*Math.sin((9\*2\*Math.PI/15)),x-20\*Math.cos((10\*2\*Math.PI/15)),y-20\*Math.sin((10\*2\*Math.PI/15)),

x-20\*Math.cos((11\*2\*Math.PI/15)),y-20\*Math.sin((11\*2\*Math.PI/15)),x-20\*Math.cos((12\*2\*Math.PI/15)),y-20\*Math.sin((12\*2\*Math.PI/15)),

x-20\*Math.cos((13\*2\*Math.PI/15)),y-20\*Math.sin((13\*2\*Math.PI/15)),x-20\*Math.cos((14\*2\*Math.PI/15)),y-20\*Math.sin((14\*2\*Math.PI/15)),

x-20\*Math.cos((15\*2\*Math.PI/15)),y-20\*Math.sin((15\*2\*Math.PI/15)),x-20\*Math.cos((16\*2\*Math.PI/15)),y-20\*Math.sin((16\*2\*Math.PI/15))

);

graphics.strokePoly(x-20\*Math.cos((2\*2\*Math.PI/15)),y-20\*Math.sin((2\*2\*Math.PI/15)),x-20\*Math.cos((2\*2\*Math.PI/15)),y-20\*Math.sin((2\*2\*Math.PI/15)),

x-20\*Math.cos((3\*2\*Math.PI/15)),y-20\*Math.sin((3\*2\*Math.PI/15)),x-20\*Math.cos((4\*2\*Math.PI/15)),y-20\*Math.sin((4\*2\*Math.PI/15)),

x-20\*Math.cos((5\*2\*Math.PI/15)),y-20\*Math.sin((5\*2\*Math.PI/15)),x-20\*Math.cos((6\*2\*Math.PI/15)),y-20\*Math.sin((6\*2\*Math.PI/15)),

x-20\*Math.cos((7\*2\*Math.PI/15)),y-20\*Math.sin((7\*2\*Math.PI/15)),x-20\*Math.cos((8\*2\*Math.PI/15)),y-20\*Math.sin((8\*2\*Math.PI/15)),

x-20\*Math.cos((9\*2\*Math.PI/15)),y-20\*Math.sin((9\*2\*Math.PI/15)),x-20\*Math.cos((10\*2\*Math.PI/15)),y-20\*Math.sin((10\*2\*Math.PI/15)),

x-20\*Math.cos((11\*2\*Math.PI/15)),y-20\*Math.sin((11\*2\*Math.PI/15)),x-20\*Math.cos((12\*2\*Math.PI/15)),y-20\*Math.sin((12\*2\*Math.PI/15)),

x-20\*Math.cos((13\*2\*Math.PI/15)),y-20\*Math.sin((13\*2\*Math.PI/15)),x-20\*Math.cos((14\*2\*Math.PI/15)),y-20\*Math.sin((14\*2\*Math.PI/15)),

x-20\*Math.cos((15\*2\*Math.PI/15)),y-20\*Math.sin((15\*2\*Math.PI/15)),x-20\*Math.cos((16\*2\*Math.PI/15)),y-20\*Math.sin((16\*2\*Math.PI/15))

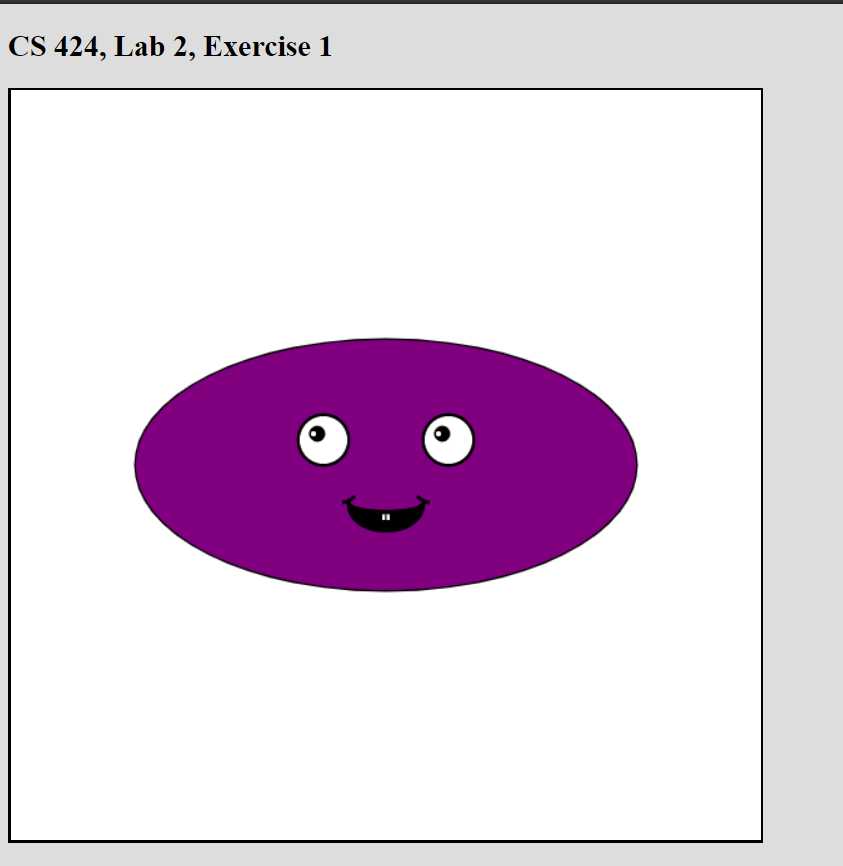
);

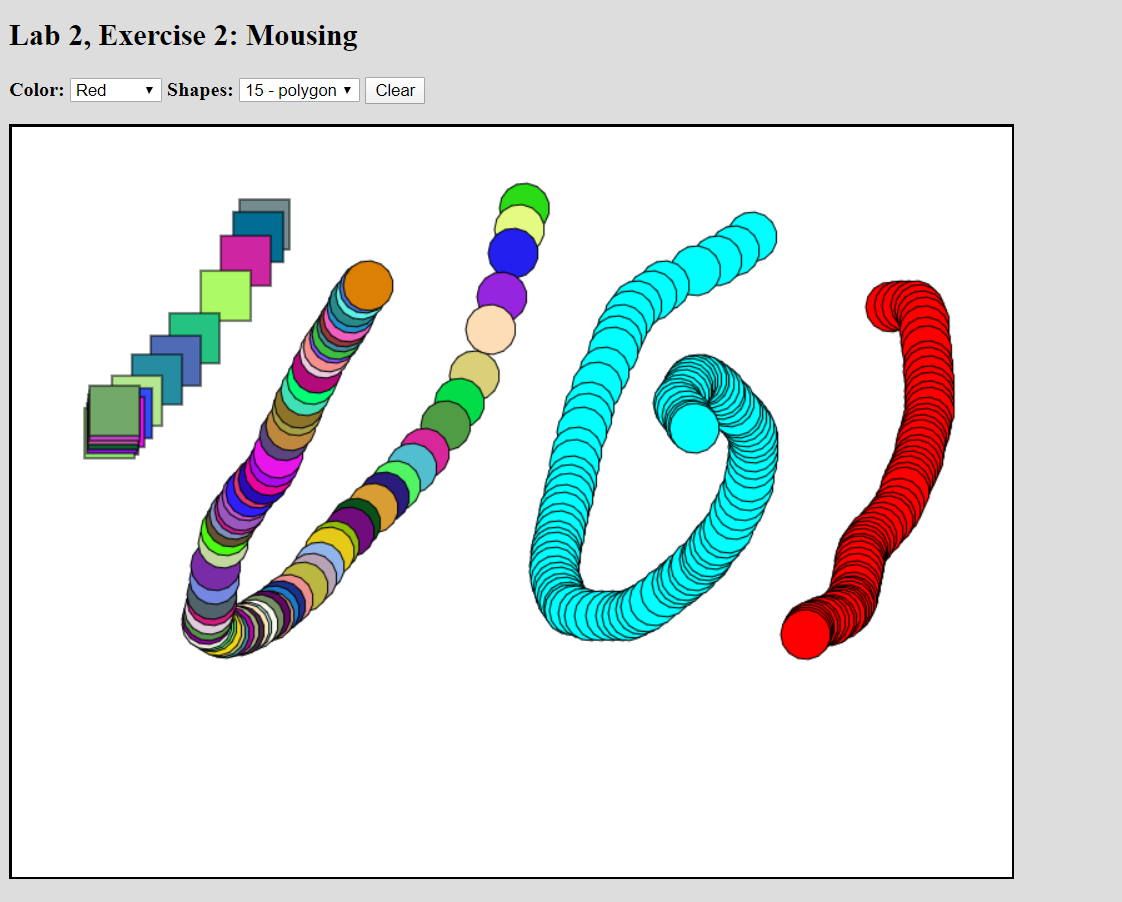
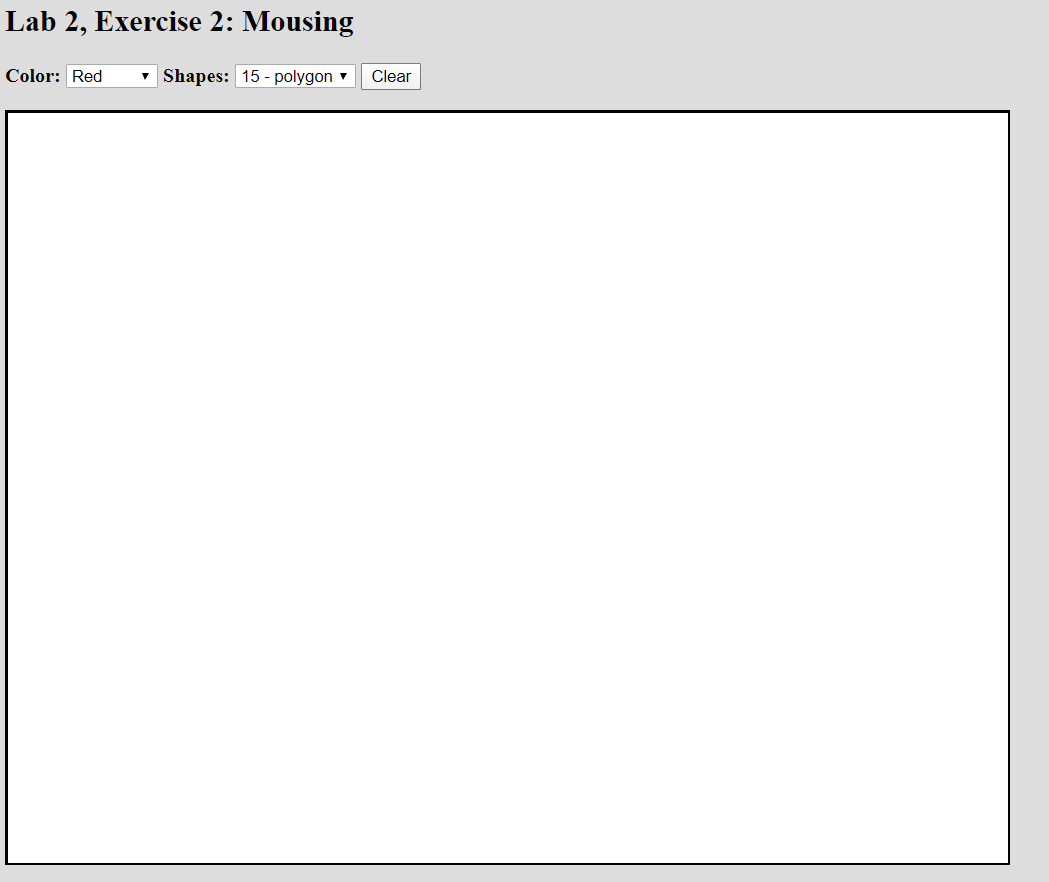
}

Link do zdalnego repozytorium zawierającego oba projekty: <https://github.com/Jarverr/GrafikaKomputerowaLab2.git>

**4. Wynik działania:**

Zrzuty ekranu prezentujące wyniki zadania 1-szego:



Zrzut ekranu prezentujący wyniki zadania 2-giego: 

**5. Wnioski:**

Na podstawie wyników uważam, że CANVAS, może okazać się bardzo przydatnym narzędziem w celu urozmaicenia i dodania prostych kształtów, możliwości graficznych na stronie. Biblioteka posiada proste komendy dzięki którym możemy rysować czasami nawet bardzo dokładne kształty, możemy dowolnie manipulować kolorami, efektami związanymi z obrazem jest to wygodne i nie wymaga skomplikowanej obsługi.